ePaga – Electronic Payment System

João António Frade Macedo de Almeida
Instituto Superior Técnico
joao.macedo@ist.utl.pt

Abstract

The possibility of using mobile devices for payments promises greater speed, convenience and ubiquity when compared with current payment methods. Hence the large potential demonstrated by the mobile payments market. However, the lack of interoperability between payment systems, along with the absence of suitable technologies, has delayed the progress in the field of mobile payments.

This project's goal is to develop a support system for electronic payments, with a special focus on mobile payments, which takes advantage of multiple payment protocols installed on the same device. The application derived by this multiprotocol support achieves higher universality, efficiency and interoperability, when compared to a single protocol application. The proposed system is based on NFC (Near Field Communication) technology. This document also describes the prototype developed with the purpose of demonstrating the system's behavior, as well as its evaluation.

Keywords: mobile computing, electronic payments, mobile payments, NFC.

1 Introduction

Over the last decade several initiatives have been developed with the aim of exploring the potential of electronic payments on mobile devices. However, only a small fraction of the projects subsists with some success. Among the wide range of reasons that lead to failure of a system include the lack of universality and interoperability with other systems, poor perception of safety and high cost of compatible equipment. In response to the high amount of projects on the market, the solution proposed in this paper presents an infrastructure that adapts itself to this heterogeneous environment. This solution aims to develop an application that supports multiple payment systems. When making a payment, the proposed solution must choose the most appropriate system, depending on the systems supported by the merchant and customer requirements.

The described solution has the following goals:

- Facilitate the development and maximize the success of mobile payment systems.
- Facilitate and unify the users access to mobile payments.

The proposed application implies the requirements described below.

- Simplicity: The system must be easy to use by customers and merchants.
- Universality: The system should support as many uses as possible in the most diverse environments, including different types of stakeholders and payments of various amounts. These uses include, for example, P2P payments and offline payments.
- Interoperability: The system should not be limited by a brand of devices, by a bank or a telecom operator. The interaction with other electronic payment systems should be facilitated, as well as the integration with traditional payment systems.
- Consistency: Despite operating in a heterogeneous and unstable environment, the system must provide a consistent interface to the user and to the systems that it supports.

A payment system includes additional requirements such as speed, cost or security. It is not up to the proposed application to ensure that these requirements are met by all the protocols, it's only required that it does not hamper the fulfillment of these requirements on the part of each protocol.

2 Actors

One of the reasons that hinder the creation of standards and reduce the success of initiatives in the area of mobile payments is the diversity of entities involved, each with different perspectives and goals. The following are the most important [1]: customers, merchants, banks, telecom operators, mobile devices manufacturers and governmental or regulatory entities.

Customers [2]: For a mobile payments service to be adopted by users it needs to differentiate itself from the current methods of payment such as cash, credit cards or cheques. The main factors that influence the customer's opinion towards a mobile payment system are ease of use, perceived security and cost.

Merchants: In addition to sharing with customers concerns like cost and security, the points that are most relevant for the merchants are the speed of the transaction and ease of integration with existing payment systems.

Banks: For banks, mobile payments represent an opportunity to offer a new service, attracting new clients,

increasing customer loyalty and maximizing profit per customer. The banks want to have control over the payment application and would like the system to be independent from telecom operators.

Telecom Operators: mobile network operators, like banks, also see mobile payments as a possibility to offer a new service, with the advantages mentioned above. Any service that requires of communication through its network represents an extra income for operators. Like banks, operators also seek control over payment applications, as well as system independence from banks.

Mobile Devices Manufacturers: Manufacturers influence the development of mobile payment systems through the introduction of new technologies in the devices. The attributes seen as favorable, in a mobile payments service, are the choice of an inexpensive technology and low time-to-market.

Governmental or Regulatory Entities: These institutions contribute by developing favorable legislation, promoting the development of standards and implementing initiatives such as the creation of a PKI (Public Key Infrastructure), assigning keys and certificates to citizens. Government agencies want to be able to, in the context of a criminal investigation, access information concerning transactions made by an individual.

3 Types of Mobile Payment Systems

Despite the high number and variety of systems developed in the mobile payments field, these can be characterized into categories representing their key attributes. These categories are listed below.

- Transaction Value [2]: the relevance of requirements such as speed, cost per transaction and security depends on the value of the transaction.
- Interaction Type [4]: a payment can be made remotely or in close proximity. These types of interactions involve different usage scenarios. The remote payments include scenarios such as the virtual point of sale or transfer C2C (Customer to Customer). Situations such as point of sale (POS), P2M (Person to Machine) or P2P (Peer to Peer) are some examples of proximity payments.
- Time of Payment [4]: the moment at which transactions are collected affects variables such as client balance control or the communication required to execute a payment. A system may be prepaid, postpaid or real-time.
- Transaction Type: at the heart of an electronic payment service is the concept of transaction. A transaction can be represented by the signing of a document, like the use of cheques. This type of system is called account-based [5]. A transaction may also be represented by the exchange of objects created by a trusted entity, similar to the use of cash. These systems are categorized as token-based [2].
- Need for Intermediaries [6]: proximity payments can require interaction with a central entity. These transactions can be categorized as online. Transactions in which this interaction is not required are called offline. The offline transactions are quicker and cheaper. They may also be executed outside mobile telephone network range. However, the lack of control by the central body during the transaction causes other problems. In account-based systems, offline payments lead to the difficulty of detecting the improper reuse of tokens, which translates into an improper multiplication of money.

4 NFC

NFC (Near Field Communication) [7] is a relatively new technology based on RFID (Radio Frequency IDentification) and compatible with it. NFC operates at 13.56 MHz and allows bit rates up to 424 Kbit/s. An NFC device can operate in three modes: P2P mode to communicate with another NFC device, as a NFC tag reader, or in NFC tag simulation mode, in which a reader sees it as a contactless card. An NFC compatible device must include the following components: an NFC antenna, an NFC chip and a secure element. The secure element has the ability to store data and run applications securely.

In addition to displaying low energy consumption, the main feature of NFC is the reduced range (3 to 30cm). This factor makes intrusions very difficult, which in turn makes protocols such as Bluetooth pairing unnecessary. Thus, the establishment of a connection is simpler and faster [8].

5 fairCASH

fairCASH [9] is a token-based and pre-paid payment system that allows remotely and offline proximity payments. The tokens of the system are transferable, that is, they can be traded between clients repeatedly before being deposited. The system allows unbreakable anonymity of the customer, preventing a client application from being associated with its owner. The customer does not need to provide personal data to the system through any kind of registration. To load the application with tokens, the client performs a bank transfer to the system for the desired amount. To reduce the risk of token duplication, each client keeps a record of received tokens. With the client's permission, this record is used by the system to calculate the origin of a duplication of tokens. The system also mentions a maximum number of times a token can be used. Each entity is identified by its digital certificate. To ensure the authenticity of tokens, they are signed by the issuing entity.

fairCASH was chosen to provide a basis for the implemented token-based protocol. This system was chosen

because of the simplicity of implementation of the described techniques.

6 System proposed by Hassinen et al.

The system proposed by Hassinen et al. in [10] takes advantage of FINE-ID, a PKI (Public Key Infrastructure) implemented at national level in Finland. This initiative assigns a pair of keys and a digital certificate to each citizen, which simplifies the implementation of systems which use asymmetric encryption, such as the solution proposed by Hassinen et al.

This solution offers two payment protocols. One of the protocols allows virtual POS payments, while the other protocol supports POS payments online, with a focus on P2M payments. This system was used as a basis for an implemented account-based protocol.

7 Architecture

The mobile payments support system ePaga intends to include all the described types of payments. Regarding short range communication technologies, the technology NFC was chosen, for being the only one that does not limit the system in terms of safety and usability.

Figure 1 represents the client's application architecture, consisting of two parts. The main part of the client application is found in the phone's operating system like Android, Symbian or iOS. This section is independent from the installed protocols. The other part of the application is found in the device's secure element, isolated from the rest of the device.

The top layer implements the interaction with the user so that the application provides a consistent user interface. The layer immediately below houses the various payment protocols. This is the only part of the architecture which varies between payment systems. The middleware of the system aims to provide an abstraction layer to the protocols that are implemented payment over it. The middleware should also, at the moment of receiving or making a payment, choose the upper layer protocol to be used. The lower layer blocks represent the features offered by the device, which are used by the middleware.

The middleware layer is divided into the following modules:

- Operations Execution Mechanisms: manage part of the operations execution which is same between protocols. They are part of the main fragment of the application.
- Protocol Selection Mechanisms: choose which of the protocols stored in the Protocols Registry that are eligible for a payment. These mechanisms use a Protocol Selection Policy to sort the protocols. They belong to the secure element.
- Protocols Registry: manages the information about the payment protocols installed on the device.
- Service Storage: maintains information about the service provided by the device. This component is used only when the device is receiving a payment. It is part of the secure element.
- Protocol Selection Policy: Selection Policy Protocol: A set of rules that, given a set of protocols and context information, orders the set of protocols according to a certain criterion, which should be configurable. It is part of the secure element.
- Local Communication: manages NFC communication with other nearby devices. Also manages the local access to the secure element of the device. It is part of the main fragment of the application.
- Remote Communication: Manage remote communication based on web services. It is part of the main fragment of the application.

7.1 Components

The ePaga system defines five entities that interact in payment transactions: client device, internal secure element, external secure element, remote merchant and remote service provider. These components and the way they communicate are depicted in Figure 2.

- Client device: houses the main fragment of client application. Together with the internal secure element

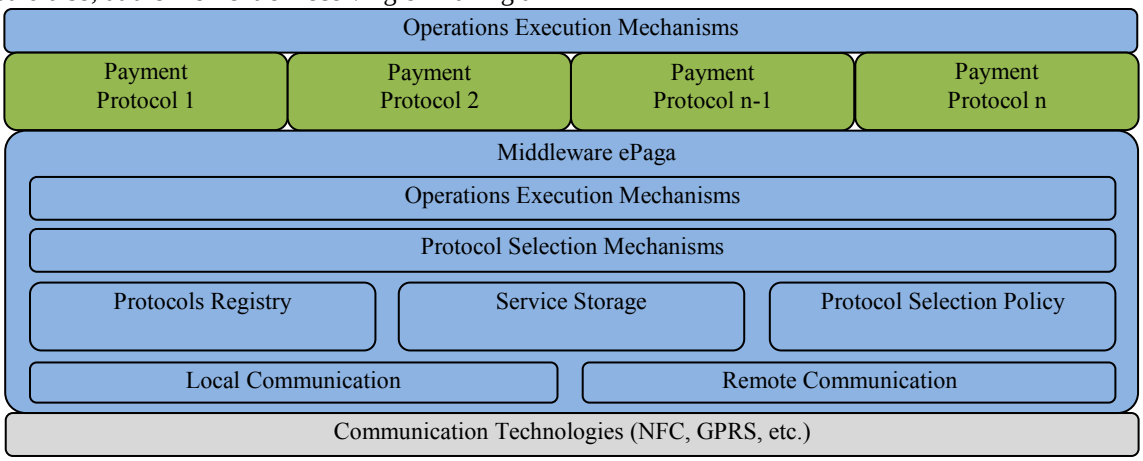

Figure 1. Software architecture of the ePaga system.

constitutes the client application. This application interacts with the client and the other four components of the system. Thus, it works as an intermediary between the entities of the system but does not contain the logic of the system. Maintains only soft-state, so this component can be replaced by an equivalent one without invalidating the client application.

- Internal secure element: hosts the critical part of the client application, including payment protocols and selection policies.
- External secure element: constitutes the receiver in a local transaction. May represent the merchant device or, in a P2P transaction, the other customer device.
- Remote merchant: represents the merchant's device in a remote transaction. Regarding the implementation of this component, the system only defines an interface that needs to be followed, so that the client application may access any remote merchant's device the same way. The remaining details of its architecture vary between protocol implementations.
- Remote service provider: server that represents the central body of the system. Just like for the remote merchant's device, the system only defines an interface to be shared between providers.

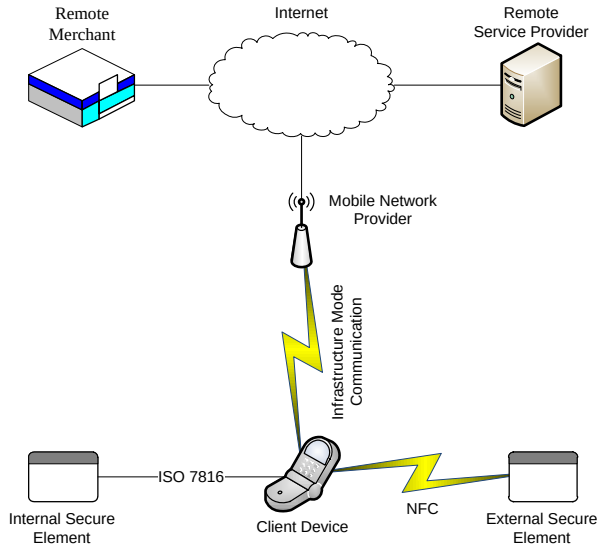

Figure 2. Simplified network diagram that illustrates the main components of the ePaga system architecture.

7.2 Operations

In the survey carried out on electronic payment systems, there were identified four operations which these systems offer their users. ePaga must include these operations so as to maximize the number of supported payment protocols. The system architecture defines the following operations: withdrawal, payment, balance checking and deposit. The payment operation can be further divided into local payment, remote payment and receipt of payment.

7.2.1 Withdrawal

The withdrawal operation represents the "recharging" of a payment system, for the protocols that include a withdrawal phase.

Depending on the protocol, the execution of this operation by the mobile device may need to be complemented by actions in another system. It's possible, for example, to develop a protocol payment where the customer, to use the system, must transfer beforehand the amount which he wants to use by other means such as ATM. The client would then perform the withdrawal operation on the mobile device, which would cause a balance synchronization of the on the device.

On the other hand, it is also possible to develop a protocol that, during the mobile device's withdrawal operation, triggers all the necessary actions to update the balance of the user. These actions may materialize themselves, for example, in an update of the value to be paid at the end of the month (on post-paid systems), or in executing a bank transfer (on real-time systems).

7.2.2 Payment

The payment operation is the main action of a payment system and represents a transfer of credits between two entities. These credits may or may not have a relationship with a monetary unit. As mentioned, the entities involved may vary between payments. The payee can be a merchant or a customer. He may be near the payer, resulting in the use of NFC to connect the payer and payee devices.

The payer and the payee may be in a different physical space, in which case the payer's mobile device uses the remote communication technology available (GSM, UMTS, LTE, etc.).

7.2.3 Deposit

The deposit operation allows the system users (customers and merchants) to convert credits from a payment protocol to credits from an external system. This operation can be used, for example, so a payee's device can notify a bank about the payments he received. It is possible to develop payment protocols that do not implement the deposit operation, if it is not necessary for the protocol execution. This is usually the case for account-based protocols. In the previous example, if the protocol informed the bank whenever a payment was made, the deposit operation could be eliminated.

7.2.4 Balance checking

The balance checking operation is not required for the execution of payment protocols. Its purpose is to show the user the balance of the installed protocols.

8 Implementation

To demonstrate the architecture of the proposed solution, an ePaga proof of concept was developed. This implementation is comprised by the application containing the proposed middleware and two test protocols.

8.1 JavaME Component

The Java ME (Micro Edition) component contains the presentation layer, the communication modules and part of the middleware. It acts as intermediary between the user, the Java Card module and server component. This component is divided into four layers: execution, presentation, smartcard and web services.

The presentation layer is responsible for interacting with the user. The user requests are forwarded to the execution layer. The main challenge of this layer is to maintain a constant interface across the different protocols. To achieve this goal, the code of this layer should be as generic as possible. Where a complete protocol abstraction is not possible, the presentation layer uses the metadata provided by the execution layer to interact with the user in a manner that makes sense for the protocol in use.

The execution layer is the brains of the midlet. Its function is to collect data from the web services and smartcard layers, in order to process commands from the presentation layer.

The smartcard and web services layers materialize the decisions taken by the execution layer.

The smartcard layer acts as an abstraction to contactless smartcard access. Communication with the Java Card module belonging to either the same device, or other NFC device, uses an ISO14443 connection.

The application's remote communication was implemented using web services. The stubs needed to access the web services were generated by the Stub Generator tool included in the Sun Java Wireless Toolkit. The toolkit version used was the 2.5.1.

8.2 Java Card Component

The Java Card component contains the critical part of the client application, namely the protocol selection logic, the storage of metadata and the payment protocols. The system relies on the secure execution and intrusion protected storage of this component. The component is divided into two layers: middleware and protocols.

The middleware layer makes up the structure that allows the application to take advantage of payment protocols. Among its functions are metadata storage and protocol selection.

The protocols used by the system are implemented in the protocols layer. Each protocol includes a payment applet that implements the logic of the protocol. In order to force the normal sequence of commands which represents the payment protocol, the applet maintains the state of the current operation. Commands outside the normal sequence of the protocol are rejected. The developed application provides base classes that simplify the implementation of protocols, by eliminating duplicate code.

The first of the payment protocols, which exemplifies a token-based protocol, allows offline payments in close proximity, especially P2P and POS transactions.

The second protocol represents the account-based protocols for virtual POS payments. This protocol is based on the protocol proposed by Hassinen et al. for virtual POS.

8.3 Remote Components

As mentioned previously, the system ePaga defines interfaces for the remote merchant and server. Since they are generic interfaces common to all protocols payments, they must be as broad as possible. Thus, all the methods define as data input and output a sequence of base64 encoded bytes. In this way any type of data used by the protocols is supported. In addition to the basic methods expected in a protocol, the interfaces of both components define an extra method, whose behavior is undefined. Thus, if the methods defined are not sufficient to predict the behavior of the protocol, the protocol implements this method.

The remote merchant and server components have been implemented in version 1.6 of the Java language. For cryptographic functions it was used the 1.46 version of the library Bouncy Castle.

8.4 Development Environment

For the development of the Java ME module is was chosen the SDK (Software Develop-ment Kit) for Nokia 6212. This SDK was chosen because it is the most complete development solution for NFC available. The SDK includes a simulator that allows testing of the Java ME module.

The Java Card module was implemented using version 2.2.2 of the Java Card Development Kit. The tests of this module used the simulators included in Development Kit: CREF (C-language Java Card RE) and JCDWE (Java Card platform Workstation Development).

In order to test Java ME and Java Card modules together, an SDK plugin was used. From the perspective of the SDK, this plugin represents a smartcard. This smartcard can be coupled to the antenna of the simulated NFC phone, or used as the internal secure element of the phone. The plugin receives commands from the phone simulator and sends them to the Java Card simulator.

9 Evaluation

This thesis aims to create a working prototype that implements the proposed architecture. This implementation includes the development of the described middleware and the presentation layer. To verify compliance with the

targets set for the ePaga, the implementation was evaluated according to qualitative criteria, whose fulfillment cannot be quantified and quantitative criteria, whose compliance can be objectively demonstrated by testing the application.

9.1 Qualitative Evaluation

The system performance is analyzed below, according to each requirement identified in the introduction.

Consistency: This requirement was met by payment and deposit operations, since the user (payer or payee) can execute these operations without knowing which protocol was used. The operations withdrawal and balance checking did not fully achieve this requirement. The withdrawal requires the user to select the protocol to be used, while the balance checking shows the balances separated by protocol. In these operations it was decided to give more information and control to the user over the application at the expense of consistency.

Interoperability: the concentration of the protocols and application state in the secure element leads to a greater independence from the device, namely regarding the device's operating system and its programming language. The system also does not limit the control over remote components to a particular player like a bank or carrier. This distribution of responsibilities by actors is left undefined. The possibility of interaction between a device that uses the system ePaga and one which does not was also not limited by the architecture.

Universality: the architecture fulfills this requirement because the operations defined allow any of the described types of payments to be implemented. However, the proof of concept implemented does not achieve the same universality. The reason for this discrepancy is related to the time required to implement protocols for all the situations envisaged. Still, the protocols implemented demonstrate that the system can support completely different types of payment such as P2P and virtual POS.

Simplicity: although this requirement has influenced the system design and the proof of concept implementation, its compliance could not be measured objectively. The simulation environment in which the system was developed invalidates tests with real users.

9.2 Quantitative Evaluation

To analyze quantitatively the performance of the implemented proof of concept, measurements were made on three aspects: operations processing time, transmitted data and size of the applications. The operations performed in the tests consisted of a withdrawal of 50 cents, a local payment of 20 cents, a remote payment of 20 cents and a 20 cents deposit.

To examine the overhead introduced in the speed of execution of the operations of the payment protocols, the time the application spends processing each operation was measured (Figure 3). In Figure 3 it can be observed that the receipt stands out negatively, in relation to other operations, with an overhead around 50%. This result is due to two factors. The ePaga application defines two steps for this operation that the dedicated application did not include. These additional steps are primarily responsible for the disparity in the figures for this operation. The second factor is related with the operation's low execution time. The payment receipt is the operation with the shortest processing time, which causes the same overhead introduced by the middleware to seem higher in this operation than in any other.

With the purpose of adding context to these values it is interesting to combine them with one example of an existing payment protocol that has been tested in an environment closer to reality. In [10] the authors present an average of 7.5 seconds to perform the steps of a local payment, including processing times and communication. Through the sum of the local payment and payment receipt operations in Figure 45, the processing time of a local payment is obtained. From this calculation for the ePaga application and the dedicated application, one obtains a value of 30% for the middleware overhead of a local payment. If the protocol evaluated in [10] was implemented in the ePaga proof of concept application, It is estimated that the average processing time would be less than 9.8 seconds. This corresponds to the worst case scenario, which assumes a communication time of zero, and consequently an execution time equal to the processing time. This is the worst case because the overhead of 30% is applied to the processing time, which in this situation would reach its maximum value.

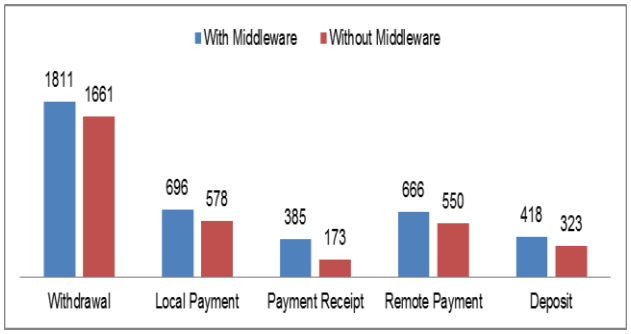

Figure 3. Graphical representation of the processing time (in milliseconds) of the main operations defined by the system.

It was also analyzed the overhead introduced in the amount of data transmitted between the mobile device and the internal and external secure elements. Between local and remote, this is the kind of communication in which middleware has a bigger impact. However, this is also the type which has less influence on the protocol performance, specifically on the cost and execution time of the transaction. The results of this analysis are illustrated in Figure 4.

Like with the processing time, the proof of concept shows an excessive value in the payment receipt operation.

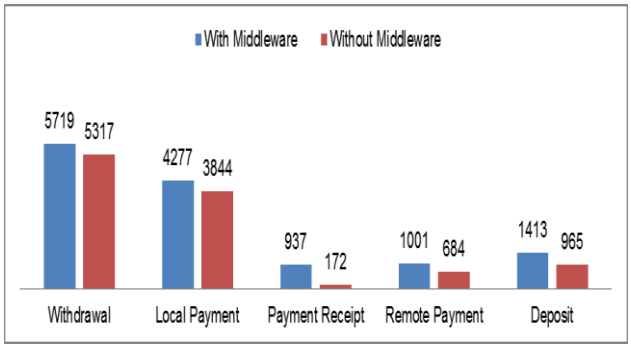

Figure 4. Graphical representation of data (in bytes) transmitted, for each operation, between the mobile device and the internal and external secure elements.

On mobile devices, the amount of data transmitted across the operator network has a direct influence on the operation cost, so this was also measured. It can be seen in Figure 5 that the difference introduced by the application ePaga is not significant. This similarity between applications is justified by the reduced data and additional steps inserted by middleware with this type of communication. In fact, the main source of communication overhead is the more inefficient encoding, caused by the use of a generic application. Dedicated applications have a greater knowledge about the types of information that the protocol produces, while the application ePaga treats data uniformly.

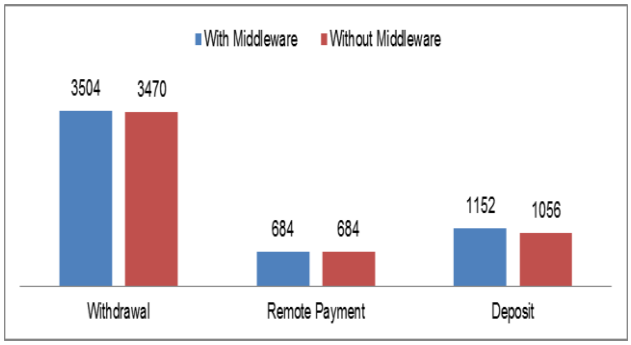

Figure 5. Graphical representation of data (in bytes) transmitted, for each operation, between the mobile device and the remote components.

In order to analyze the implementation effort required to develop the applications, the number of lines of code for each was measured. These values are illustrated in Figure 6, which shows that approximately half of the written lines are written in JavaME. On one hand this code is generally easier to write on the other hand the code is less relevant for the logic of the protocol. Thus, the use of the ePaga application and its middleware reduces the effort for implementing a protocol. It is also important to mention that the middleware offers a basis for the protocols code, which represents another contribution to simplify the code.

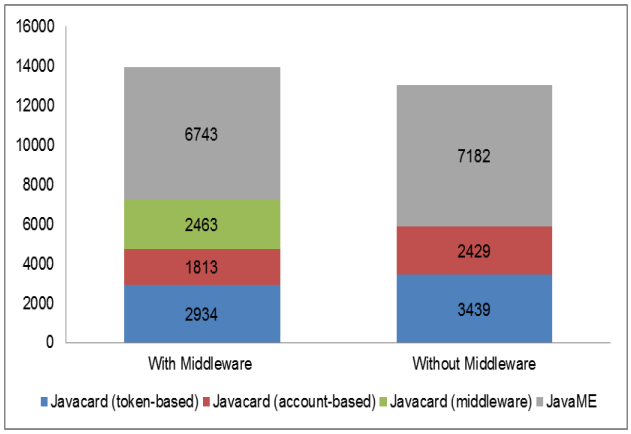

Figure 6. Graphical representation of the number of lines of code that were needed to develop the implemented applications.

Since the available memory for installing code in internal secure element is a limited resource, an analysis was made regarding the memory occupation of the implemented application. The results of this analysis are presented in Figure 7. The figures show that the memory required to install the middleware in a internal secure element is similar to that occupied by a payment protocol. From Figure 7 it is clear that the middleware reduces the space required to develop protocols, but the value of this reduction varies significantly depending on the protocol. It was also found that, as expected, this added value becomes more important as the number of protocols installed increases.

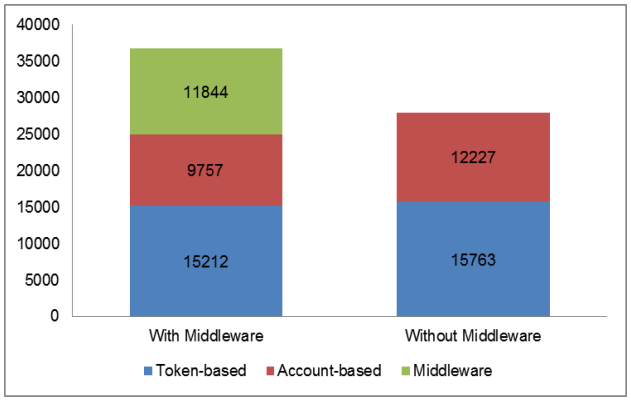

Figure 7. Graphical representation of the memory occupied by applications implemented in the internal secure element.

10 Conclusions

Mobile payments still fail to permanently conquer the market. However, the mobile payments area continues to show tremendous potential. In addition to the many existing systems, categorized and exemplified in this document there is a constant investment in new projects all over the world **Error! Reference source not found.Error! Reference source not found.**. On the other hand, the pilot tests carried out using this type of system

continue to confirm the public interest to adhere to one of the services **Error! Reference source not found.**.

This high number of initiatives, in along with the current lack of standards, allows to foresee a market filled with systems offered by different organizations, many of them mutually incompatible. In response to this scenario, a system was proposed that supports multiple protocols. This system allows the transaction participants to communicate through the most appropriate protocol, selected among the protocols supported by both.

Next are presented the contributions that the proposed system architecture brings to the ecosystem of mobile payments. Such contributions are summarized in Table 1.

- Reduces the development effort and maintenance of new services and protocols: current payment protocols are often embedded in applications that use them, mixing business logic of the service with payment of said service. The proposed architecture isolates the payment feature from the rest of the application. Thus, the implementation of new services is simplified. In addition, the payment protocols represent modules separated from the rest of the ePaga architecture. Therefore, to implement a new payment protocol it is no longer necessary to implement an application from scratch. This advantage was demonstrated in the previous chapter (Figure 6).
- Increases portability of developed protocols: the same protocol works on any operating system on which the ePaga application is implemented.
- Boosts the success of non-universal protocols: existing payment systems show the difficulty in creating a protocol that can be used in any situation. In the proposed system, several protocols of limited universality create an application with superior universality.
- Maximize the payment efficiency: even assuming that there are two protocols that can be used in any situation, it is difficult for each of them to always be the most efficient. The ePaga application chooses the most efficient protocol for each payment, reducing its average cost.
- Automatic protocol selection: As explained above, such as solutions provided by GlobalPlatform allow multiple applets to coexist in the same secure element. Thus the two preceding paragraphs could be achieved by installing several applications, each with its own protocol. However, the coexistence of payment applications would not be transparent to the user. So he would have to choose manually the protocol to use, which would become impractical. With the ePaga application, the protocol is selected automatically and transparently to the user.
- Interface independent from the chosen protocol: Another disadvantage of the use of various applications is the coupling of an interface and payment protocol. With the ePaga system, the interface is dependent on the desired action by the user, and not the protocol being used.

Table 1. Comparison between ePaga system and existing alternatives. The first column represents an application with a single payment protocol. The second column represents the use of various payment applications, each with a single protocol.

	One application	Multiple applications	ePaga
Life cycle management	Medium	Complex	*Midlets*: simple *Applets*: complex
Life cycle management with GP	Simple	*Midlets*: complex *Applets*: simple	Simple
New protocols cost	High	High	Low
Complexity of use	Low	High	Low
Payment efficiency	Low	Medium	High
Necessary universality for each Protocol	High	Low	Low

In addition, the following are conclusions from the evaluation and that derive from the requirements of payment systems.

- The system does not add a significant overhead to the cost per transaction: the cost could be increased if the system would raise the amount of data exchanged remotely during the transaction.
- The system increases the proximity transaction time: the amount of overhead must be determined from tests in an environment closer to real conditions. Still, in the example used in the evaluation, the protocol would still meet the payment requirements that were proposed (below 15 s).

With these contributions, the system ePaga, whose concept was demonstrated in this thesis aims to help explore the true potential of mobile payments.

10.1 Future Work

Listed in this section are the steps that should follow the implementation of the proof of concept described in this document.

- Tests in an environment closer to the real condictions, particularly with real mobile devices and users. These tests offer a more precise notion of system performance.
- Implementation of prototypes in other operating systems. This step exploits the portability that the proposed architecture allows.
- Implement the automatic service request described in the architecture. The interaction with web pages of merchants and NFC tags simplifies and accelerates the payment process.
- Add more protocols to the system. The inclusion of new protocols to the system allows the detection of op-

portunities to improve the architecture or implementation.

References

[1] Karnouskos, S. Mobile payment: A journey through existing procedures and standardization initiatives, Communications Surveys & Tutorials, IEEE, 6(4), 44-66, 2004

[2] N. Kreyer, K. Pousttchi, et al. Characteristics of Mobile Payment Procedures, Proceedings of the ISMIS 2002 Workshop on M-Services, 2002

[3] Andriew S. Lim, Inter-consortia battles in mobile payments standardisation, Electronic Commerce Research and Applications, 7(2), 202–213, 2008

[4] S. Karnouskos et al. Secure Mobile Payment — Architecture and Business Model of SEMOPS, Proceedings of the EURESCOM summit, 2003

[5] Xiaolin Zheng, Deren Chen, Study of Mobile Payments System, Proceedings of the IEEE International Conference on E-Commerce (CEC'03), 2003

[6] R. K. Balan, N. Ramasubbu, et al. mFerio: The design and evaluation of a peer-to-peer mobile payment system, Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services, 2009

[7] Diogo Simões, Sistema de Fidelização sobre NFC (Near Field Communication), 2008

[8] J. J. Chen, C. Adams, Short-range wireless technologies with mobile payments systems, The 6th International Conference on Electronic Commerce (ICEC), 2004

[9] H. Kreft, Cashing up with Mobile Money – the fairCASH Way, Euro mGov 2005, 2005

[10] Marko Hassinen, Konstantin Hypponen, Elena Trichina, Utilizing national public-key infrastructure in mobile payment systems, Electronic Commerce Research and Applications, Volume 7, Issue 2, Special Section: Research Advances for the Mobile Payments Arena, Summer 2008, Pages 214-231, ISSN 1567-4223